

Docket No. **302132.01**

**MSP344US**

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re **PATENT** application of:

Applicant: Kirt Debique et al.

Serial No.: 10/635,730

For: DEMULTIPLEXER APPLICATION PROGRAMMING INTERFACE

Filing Date: August 6, 2003

Examiner: Kimbleann C. Verdi

Art Unit: 2194

**AMENDMENTS PER 37 CFR 1.312 IN RESPONSE TO NOTICE OF ALLOWANCE**

**DATED SEPTEMBER 14, 2010**

Mail Stop Issue Fee  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Dear Sir:

Favorable reconsideration of the above-identified application is respectfully requested in view of the following remarks and amendments.

**AMENDMENTS**

**IN THE CLAIMS:**

*Please amend claims 1, 2, 7, 10, 14, and 16-21 as follows:*

1. (Currently Amended) A computer-readable storage medium having computer-executable instructions, that when executed on a computing system, perform a method comprising:

implementing an interface for communication with a demultiplexer object, the interface taking multiplexed multimedia data as input and outputting demultiplexed elementary media streams, the interface comprising:

an Initialize method configuring the demultiplexer object;

a GetPendingPresentationDescriptor method retrieving a ~~the~~ next pending presentation;

a SetPresentationDescriptor method dynamically setting an active presentation descriptor of the demultiplexer object to the ~~the~~ [[a]] next pending presentation when an active presentation exists only if all output associated with the active presentation has been serviced, wherein the active presentation descriptor describes media types of an elementary stream, and facilitates selection of streams to be extracted by the demultiplexer object;

wherein if the SetPresentationDescriptor method is called attempting to set the active presentation descriptor to the next pending presentation when the active presentation exists and not all output associated with the active presentation has been serviced, the SetPresentationDescriptor method indicates that the active presentation descriptor cannot be set to the next pending presentation because not all output associated with the active presentation has been serviced, wherein the SetPresentationDescriptor method includes a pointer to a presentation descriptor object;

a ProcessInput method providing a new input muxed stream to the demultiplexer object;

a ProcessOutput method retrieving at least one elementary stream from the ~~[[an]]~~ active presentation determined based on the dynamically set active presentation descriptor; and

a Flush method flushing currently queued input and output samples.

2. (Currently Amended) The computer-readable storage medium of claim 1 wherein the interface ~~further~~ comprises a GetPresentationDescriptor method retrieving a clone of the ~~currently~~ active presentation descriptor on the demultiplexer object.

3. (Previously Presented) The computer-readable storage medium of claim 2 wherein the GetPresentationDescriptor method includes a presentation descriptor.

4. (Cancelled)

5. (Previously Presented) The computer-readable storage medium of claim 1 wherein the GetPendingPresentationDescriptor method includes a pending presentation descriptor.

6. (Previously Presented) The computer-readable storage medium of claim 1 wherein the Initialize method includes parameters, the parameters comprising:

a muxed stream descriptor;

a selected media type for the muxed stream descriptor;

an array of major types of elementary streams; and

a count of major types in the array of major types.

7. (Currently Amended) A computer implemented method, comprising:

implementing, using a processor, an interface for communication with a demultiplexer object, the interface taking multiplexed multimedia data as input and outputting demultiplexed elementary media streams, the interface comprising:

- an Initialize method configuring the demultiplexer object;

- a GetPendingPresentationDescriptor method retrieving a the next pending presentation;

- a SetPresentationDescriptor method dynamically setting an active presentation descriptor of the demultiplexer object to the [[a]] next pending presentation when an active presentation exists only if all output associated with the active presentation has been serviced, wherein the active presentation descriptor describes media types of an elementary stream, and facilitates selection of streams to be extracted by the demultiplexer object;

- wherein if the SetPresentationDescriptor method is called attempting to set the active presentation descriptor to the next pending presentation when the active presentation exists and not all output associated with the active presentation has been serviced, the SetPresentationDescriptor method indicates that the active presentation descriptor cannot be set to the next pending presentation because not all output associated with the active presentation has been serviced, and wherein the SetPresentationDescriptor method includes a pointer to a presentation descriptor object;

- a ProcessInput method providing a new input muxed stream to the demultiplexer object, wherein the ProcessInput method includes a pointer to a sample object;

- a ProcessOutput method retrieving at least one elementary stream from an active presentation determined based on the dynamically set active presentation descriptor, wherein the ProcessOutput method includes a stream identifier and a pointer to a pointer to a sample object; and

- a Flush method flushing currently queued input and output samples.

8. (Previously Presented) The computer-readable storage medium of claim 1 wherein the ProcessInput method includes a pointer to a sample object.

9. (Previously Presented) The computer-readable storage medium of claim 8 wherein the ProcessInput method includes a return value having a new presentation flag.

10. (Currently Amended) The computer-readable storage medium of claim 9, having further computer executable instructions for performing the method comprising: if the new presentation flag has a TRUE value:

calling the ~~the~~ ~~[[a]]~~ GetPendingPresentationDescriptor method to retrieve the next pending presentation;

selecting desired streams; and

calling the SetPresentationDescriptor method to enable processing of samples from the demultiplexer object's input queue.

11. (Previously Presented) The computer-readable storage medium of claim 1 wherein the ProcessOutput method includes a stream identifier and a pointer to a pointer to a sample object.

12. (Previously Presented) The computer-readable storage medium of claim 11 wherein the ProcessOutput method includes an output return value.

13. (Previously Presented) The computer-readable storage medium of claim 12 wherein the output return value includes one of an end of stream error code and a no more data error code.

14. (Currently Amended) A computer-readable storage medium having computer-executable instructions, that when executed on a computing system, perform a method comprising:

implementing an interface for communication with a demultiplexer object, the interface taking multiplexed multimedia data as input and outputting demultiplexed elementary media streams, wherein the interface takes multiplexed data as an in-memory buffer of data, the interface comprising:

an Initialize method configuring the demultiplexer object;

a GetPendingPresentationDescriptor method retrieving ~~to retrieve~~ the next pending presentation;

a SetPresentationDescriptor method dynamically setting an active presentation descriptor of the demultiplexer object to a next pending presentation when an active presentation exists only if all output associated with the active presentation has been serviced, wherein if the SetPresentationDescriptor method is called attempting to set the active presentation descriptor to the next pending presentation when the active presentation exists and not all output associated with the active presentation has been serviced, the SetPresentationDescriptor method indicates that the active presentation descriptor cannot be set to the next pending presentation because not all output associated with the active presentation has been serviced, wherein the SetPresentationDescriptor method includes a pointer to a presentation descriptor object;

a ProcessInput method providing a new input muxed stream to the demultiplexer object, wherein the ProcessInput method includes a return value having a new presentation flag;

if the new presentation flag has a TRUE value:

calling the ~~the~~ [[a]] GetPendingPresentationDescriptor method to retrieve the next pending presentation;

selecting desired streams; and

calling the SetPresentationDescriptor method to enable processing of samples from the demultiplexer object's input queue;  
a ProcessOutput method retrieving at least one elementary stream from the ~~an~~ active presentation determined based on the dynamically set active presentation descriptor;  
a Flush method flushing currently queued input and output samples; and  
a GetPresentationDescriptor method to retrieve a clone of the ~~currently~~ active presentation descriptor on the demultiplexer object.

15. (Previously Presented) The computer-readable storage medium of claim 14 wherein the multiplexed data has a format comprising at least one of Digital Video, MPEG2, and ASF.

16. (Currently Amended) The computer-readable storage medium of claim 1, having ~~further~~ computer executable instructions for performing the method comprising:  
storing an Initialize data structure for use in a demultiplexer, comprising:  
a first field containing a header;  
a second field containing a muxed stream descriptor;  
a third field containing a selected media type of the muxed stream descriptor;  
a fourth field containing an array of major types of elementary streams;  
and  
a fifth field containing a count of major types in the array of major types.

17. (Currently Amended) The computer-readable storage medium of claim 1, having ~~further~~ computer executable instructions for performing the method comprising:  
storing a SetPresentationDescriptor data structure for use in a demultiplexer, comprising:

- a first field containing a header; and
- a second field containing a presentation descriptor.

18. (Currently Amended) The computer-readable storage medium of claim 1, having ~~further~~ computer executable instructions for performing the method comprising:  
storing a GetPresentationDescriptor data structure for use in a demultiplexer, comprising:

- a first field containing a header; and
- a second field containing a presentation descriptor.

19. (Currently Amended) The computer-readable storage medium of claim 1, having ~~further~~ computer executable instructions for performing the method comprising:  
storing a GetPendingPresentationDescriptor data structure for use in a demultiplexer, comprising:

- a first field containing a header; and
- a second field containing a pending presentation descriptor.

20. (Currently Amended) The computer-readable storage medium of claim 1, having ~~further~~ computer executable instructions for performing the method comprising:  
storing a ProcessInput data structure for use in a demultiplexer, comprising:

- a first field containing a header; and
- a second field containing a pointer to a sample object.



21. (Currently Amended) The computer-readable storage medium of claim 1, having ~~further~~ computer executable instructions for performing the method comprising:

storing a ProcessOutput data structure for use in a demultiplexer, comprising:

- a first field containing a header;
- a second field containing a stream identifier; and
- a third field containing a pointer to a pointer to a sample object.

**REMARKS**

Initially, the undersigned would like to thank the Examiner for her time on 12/14/10 wherein at least some of the subject matter herein was discussed over the phone. Claims 1-3 and 5-21 were indicated as allowed in the 9/14/10 Notice of Allowance, and allowance of the same is noted with appreciation. Claims 1, 2, 7, 10, 14 and 16-21 are amended herein per 37 CFR 1.312 in response to the 9/14/10 Notice of Allowance. The undersigned would like to thank Examiner Verdi for her time on or about 12/10/10 in discussing at least some of the amendments made herein over the phone.

Entry of the amendments herein is respectfully requested at least because it is believed that the amendments do not introduce new matter, but merely place the claims in a compliant format. Entry of the amendments herein is thus respectfully requested.

**CONCLUSION**

Should a telephone call be warranted to resolve any outstanding issues in this case, the Examiner (or any other relevant party) is invited to contact the undersigned at the telephone number provided below.

Should any fees be due as a result of the filing of this response, the Commissioner is hereby authorized to charge the Deposit Account Number 50-5088, **302132.01**.

Respectfully submitted,  
MICROSOFT CORPORATION

By /William J. Cooper/

William J. Cooper  
Reg. No. 44,629

Microsoft Corporation  
One Microsoft Way  
Redmond WA 98052-6399  
Direct telephone (425) 707-9382